A

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/043,809 | 01/10/2002 | Darpan Dinker | 5181-92500 | 7848 |

7590 12/20/2005

B. Noel Kivlin
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

| EXAMINER | |
|---|---|
| GAYESKI, MICHAEL R | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2143 | |

DATE MAILED: 12/20/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

|  |  |  |  |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>10/043,809 | **Applicant(s)**<br>DINKER ET AL. |  |
|  | **Examiner**<br>Michael R. Gayeski | **Art Unit**<br>2143 |  |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *10 January 2002*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 January 2002* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *2/17/2005*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

## DETAILED ACTION

1. Claims 1-34 are pending in the instant application.

### *Claim Rejections - 35 USC § 112*

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim **2** is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

4. **Claim 2** recites the limitation "the manager service" in lines 7 and 9. It is not particularly clear as to which manager service the claim refers to. To expedite the examination process, the limitation will hereinafter be interpreted as referring to the manager service on the first (non-backup) server.

### *Claim Rejections - 35 USC § 102*

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims **1, 7-22, 24, and 26-29** are rejected under 35 U.S.C. 102(b) as being anticipated by Devarakonda et al. (U.S. Patent 5,454,108) (hereinafter 'Devarakonda').

7. With regard to **claim 1**, Devarakonda teaches the substantive limitations of the claim, including a server computer comprising:

a first memory storing program instructions, a first processor coupled to the first memory, and wherein the first processor is operable to execute program instructions stored in the first memory to implement a manager service **[See Figure 1, items 112 and 114(1)-114(3); and Column 3, lines 8-53: The local lock server and lock control servers implement the manager service, while a memory, processor, and processor executing instructions from memory are inherent to computer-based systems]**;

wherein the manager service is operable to receive the requests sent by the plurality of processes executing on the plurality of client computers, wherein each request includes a request to acquire access to a data object from a plurality of data objects **[See Figure 3; and Column 5, lines 51-60: The client processes are process contacting the LLM for read/write privileges]**;

wherein the manager service is operable to respond to the requests by coordinating access rights for the plurality of data objects such that, at any given time, one of the following conditions is met for each data object:

a) One or more client processes currently have read access rights to the data object and no client processes currently have write access rights to the data object **[See Figures 6A-6E; and Column 7, lines 41-65: A read token always conflicts with a write token]**; or

b) One client process currently has write access rights to the data object and no other client process currently has read or write access rights to the data object **[See Figures 6D - 6E; and Column 8, lines 1-19: A write token conflicts with all other tokens]**.

8. With regard to **claim 7**, Devarakonda further teaches the plurality of data objects are stored on one of the client computers **[See Column 3, lines 16-20: The system is peer-to-peer, so each node serves as a client or a server, depending on the taken action]**.

9. With regard to **claim 8**, it presents no substantive limitations above those of claim 1, and is rejected for similar reasons.

10. With regard to **claim 9**, Devarakonda further teaches that the manager service executes on one of the client computers **[See Column 3, lines 16-20]**.

11. With regard to **claims 10, 11, and 12**, Devarakonda further teaches a server computer coupled to the client computers, wherein the manager service executes on the server computer; wherein the first computer is the server computer; and wherein the first computer is one of the client computers **[See Figure 1; and Column 3, lines 16-20: The system is peer-to-peer, so each node serves as a client or a server, depending on the desired operation]**;

12. With regard to **claim 13**, it presents no substantive limitations above those of claim 1, and is rejected for similar reasons.

13. With regard to **claim 14**, Devarakonda further teaches said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire read access rights for reading from the data object **[See Figures 6A-6C; and Column 7, lines 51-66]**;

and wherein said first client computer accessing the data object comprises the first client computer reading from the data object **[See Figure 6B, item 904(1)]** .

14. With regard to **claim 15**, Devarakonda further teaches that said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire write access rights for writing to the data object **[See Figures 6D and 6E]**;

and wherein said first client computer accessing the data object comprises the first client computer writing to the data object **[See Figure 6E, item 904(1)]**.

15. With regard to **claim 16**, Devarakonda further teaches the first client computer sending a second request to the server computer to release access rights for the data object **[See Figure 7: "Acquire token"]**;

and the server computer reclaiming the access rights in response to the second request **[See Figure 7: "Send revoke/downgrade to other nodes"]**.

16. With regard to **claim 17**, Devarakonda further teaches:

wherein said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire write access rights for writing to the data object **[See Figure 6D]**;

wherein said server computer granting the access rights to the first client computer in response to the first request comprises the server computer granting both read and write access rights for the data object to the first client computer **[The write lock disclosed by Devarakonda is an exclusive ("upgraded") lock. Thus, a write lock provides permission to read]**;

wherein said first client computer sending a second request to the server computer to release access rights for the data object comprises the first client computer sending a request to the server computer to release write access rights for the data object **[See Column 5, line 51 – Column 6, line 66: When a client process requests a read operation, the first LLM determines if any other LLMs hold a conflicting write lock. If one does, the first LLM requests that the other LLMs release the lock]** ;

wherein said server computer reclaiming the access rights in response to the second request comprises the server computer reclaiming the write access rights; and wherein the first client computer still holds read access rights for the data object after said server computer reclaiming the write access rights **[See Figure 7; and Column 9, line 63 – Column 10, line 8]**.

17. With regard to **claim 18**, Devarakonda further teaches the first client computer sending a third request to the server computer to release read access rights for the data object **[See Column 6, lines 48-57]**;

and the server computer reclaiming the read access rights in response to the third request **[See Column 6, lines 58-66]**.

18. With regard to **claims 19, 20, and 21**, the system of Devarakonda is peer-to-peer, so each node functions as both a client and a server, depending on the particular action taken. Thus, the data objects are stored on both the client and the server.

19. With regard to **claim 22**, Devarakonda further teaches the server computer receiving a request to begin coordinating access rights for the data object, before said first client computer sending the first request to the server computer to acquire access rights for accessing the data object **[See Column 4, lines 15-26]**;

and the server computer maintaining state information regarding which client computers hold access rights for the data object in response to said receiving the request to begin coordinating access rights **[See Column 3, line 36 – Column 4, line 3: The LLM includes a lock state information]**.

20. With regard to **claim 24**, Devarakonda further teaches said first client computer accessing the data object comprises the first client computer calling an application programming interface (API) for accessing the data object **[See Column 2, lines 47-52]**;

wherein the API checks to ensure that the first client computer holds access rights for the data object before allowing the first client computer to access the data object **[See Column 2, lines 53-64]**.

21. With regard to **claim 26**, Devarakonda further teaches said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire read access rights for reading from the data object **[See Figure 6A; and Column 7, lines 41-50]**;

wherein said server computer granting the access rights to the first client computer comprises the server computer granting read access rights to the first client computer **[See Figures 6A and 6B]**;

wherein the method further comprises: a second client computer sending a second request to the server computer to acquire read access rights for reading from the data object **[See Figure 6B; and Column 7, lines 51 – 65]**;

and the server computer granting read access rights to the second client computer in response to the second request **[See Figure 6C; and Column 7, lines 51 – 65]**.

22. With regard to **claim 27**, Devarakonda further teaches said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire read access rights for reading from the data object **[See Figure 6A; and Column 7, lines 41-50]**;

wherein said server computer granting the access rights to the first client computer comprises the server computer granting read access rights to the first client computer **[See Figures 6B and 6C; and Column 7, lines 51 – 65]**;;

wherein the method further comprises a second client computer sending a second request to the server computer to acquire write access rights for writing to the data object **[See Figure 6D; and Column 7, line 67 – Column 8, line 19]**; and

wherein the server computer does not grant write access rights to the second client computer until the first client computer releases its read access rights for the data object **[See Figure 6E; and Column 7, line 67 – Column 8, line 19]**.

23. With regard to **claim 28**, Devarakonda further teaches said first client computer sending the first request to the server computer comprises the first client computer sending a request to the server computer to acquire write access rights for writing to the data object **[See Figure 6D; and Column 7, line 67 – Column 8, line 19]**;

wherein said server computer granting the access rights to the first client computer comprises the server computer granting write access rights to the first client computer **[See Figure 6E; and Column 7, line 67 – Column 8, line 19]**;

wherein the method further comprises a second client computer sending a second request to the server computer to acquire access rights for accessing the data object **[See Figure 7: "revoke/downgrade"]**; and

wherein the server computer does not grant access rights to the second client computer until the first client computer releases its write access rights for the data object **[See Figure 7; and Column 9, line 63 – Column 10, line 8: When other clients have a conflicting write token, the other clients are forced to downgrade or release the token]**.

24. With regard to **claim 29**, Devarakonda further teaches:

a second client computer sending a second request to the server computer to acquire access rights for accessing the data object **[See Figure 6D; and Column 7, line 67 – Column 8, line 19]**

the server computer reclaiming the access rights from the first client computer in response to the second request **[See Figure 7; and Column 9, line 63 – Column 10, line 8]**;

the server computer granting access rights to the second client computer in response to the second request **[See Figure 7; and Column 9, line 63 – Column 10, line 8**;

and the second client computer accessing the data object **[See Column 3, lines 36-51: The client processes read and write to the file]**.

### *Claim Rejections - 35 USC § 103*

25. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

26. Claims **5, 6, 30, 31, and 32** are rejected under 35 U.S.C. 103(a) as being unpatentable over Devarakonda in view of Wikstrom et al. (U.S. Patent 5,913,213) (hereinafter 'Wikstrom').

27. With regard to **claim 5**, Devarakonda teaches the substantive limitations of the base claims, and further teaches that the manager service is operable to grant to a first client process access rights for a first data object in a first mode **[See Devarakonda, Page 294, Figures 2 and 3]**. Devarakonda is silent, however, on implementing a 'lazy locking' mechanism, wherein said granting the access rights in the first mode comprises granting the access rights such that the first client process is not required to communicate with the manager service to release the access rights. Rather, the locking method taught by Devarakonda always requires communications with either the LLM or LCS.

Wikstrom, in the same field of endeavor, teaches a locking mechanism wherein a local lock flag is kept at a entity (process), and wherein the entity checks the local flag first, and, if it is not set, the entity performs the operation without contacting the lock manager **[See Wikstrom, Abstract: "[T]hereby enabling a successive access [...]"]**.

It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to implement a local lock flag as taught by Wikstrom in the processes of Devarakonda, with the motivation of reducing the overhead associated with obtaining a lock **[See Wikstrom, Column 2, lines 36-49]** Thus, claim 5 is rejected.

28. With regard to **claim 6**, the combination of Devarakonda and Wikstrom discussed immediately above (hereinafter 'Devarakonda-Wikstrom') further teaches wherein the manager service is operable to communicate with the first client process to reclaim the access rights in response to a request from a second client process to acquire access rights for the first data object **[See Column 7, line 67 – Column 8, line 19: Nodes pass messages to other nodes]**.

29. With regard to **claim 30**, Devarakonda-Wikstrom additionally teaches:

a first process executing on a first client computer calling a first method to acquire access rights for accessing the data object **[See Devarakonda, Column 2, lines 47-52; and Column 4, lines 15-26]**;

the first client computer communicating with a server computer to acquire the access rights in response to said first process calling the first method **[See Devarakonda, Column 4, lines 28-33]**;

the first process executing on the first client computer accessing the data object **[See Devarakonda, Column 4, lines 28-33]**;

the first process executing on a first client computer calling a second method to release the access rights **[See Devarakonda, Column 8, line 63 – Column 8, line 7]**;

and the first client computer releasing the access rights in response to said first process calling the second method, wherein said releasing the access rights does not comprise the first client computer communicating with the server computer **[See Wikstrom, Abstract]**

30. With regard to **claim 31**, Devarakonda-Wikstrom further teaches:

the first process executing on the first client computer calling the first method again to re-acquire access rights for accessing the data object **[See Wikstrom, Abstract]**;

and the first client computer granting the access rights to the first process without communicating with the server computer **[See Wikstrom, Abstract]**.

31. With regard to **claim 32**, Devarakonda-Wikstrom further teaches:

a second process executing on a second client computer calling the first method to acquire access rights for accessing the data object, after said first process calling the second method to release the access rights **[See Devarakonda, Column 2, lines 47-52; and Column 5, line 51 – Column 6, line 7]**;

the second client computer communicating with the server computer to acquire the access rights in response to said second process calling the first method **[See Devarakonda, Column 2, lines 47-52; and Column 5, line 51 – Column 6, line 34]**;

and the server computer communicating with the first client computer to reclaim the access rights in response to said second client computer communicating with the server computer to acquire the access rights **[See Figure 7: "Send revoke/downgrade to other nodes"]**.

32. Claims **4, 25, and 33** are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Patent Application Publication 2002/0143958) (hereinafter 'Montero') in view of Wikstrom.

33. With regard to **claim 33**, Montero, in the analogous art of computer networking, teaches the substantive limitations of the claim, including:

a client computer operable to transmit HTTP requests **[See Figure 1, item 16]**;

a web server computer coupled to the client computer **[See Figure 1, item 13]**;

a plurality of application server computers coupled to the web server computer **[See Figure 1, items $14_1$, $14_2$, ... $14_n$]**;

wherein the web server computer is operable to receive HTTP requests from the client computer and distribute the requests among the application server computers **[See Paragraphs 34 and 35]**;

wherein each application server computer is operable to respond to a request received from the web server computer by:

accessing the HTTP session data to process the request **[See Paragraphs 35 and 36: The session data is stored in the back-end databases]**;

Montero is silent, however, on including a manager service to grant access rights to HTTP session data, or that the application server is operable to respond to a request by communicating with the manager service to release the access rights for the HTTP session data for the client computer.

Wikstrom, in analogous art, teaches a manager service granting read or write access rights to resources **[See Wikstrom, Abstract]**. Combining the teachings by replacing the back end database **(18)** of Montero with the distributed database including a distributed lock manager as taught by Wikstrom would yield the invention as claimed. It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to perform such a combination, with the motivation of increasing the performance of the database while reducing the locking overhead normally associated with distributed databases. Thus, claim 30 is rejected.

34. With regard to **claim 4**, the combination of Montero and Wikstrom discussed immediately above (hereinafter Montero-Wikstrom) teaches that HTTP session data is stored in the databases **[See Montero, Paragraphs 19 and 20]**.

With regard to **claim 25**, Montero-Wikstrom teaches that the data structure storing the session data is a Java object, specifically a Javax.servlet.http.HTTPsession object, which is subsequently stored in the database **[See Montero, Paragraphs 12 and 35-38]**, and an HTTPSession object as defined by the Java specifications inherently contains callable methods. Accordingly, access rights for an executable component having one or more methods would be obtained before the client would be allowed to retrieve and execute said methods.

35. Claim **34** is rejected under 35 U.S.C. 103(a) as being unpatentable over Montero in view of Wikstrom as applied to claim 33 above, and further in view of Devarakonda as applied to claim 5 above.

36. With regard to **claim 34**, Montero-Wikstrom teaches the substantive limitations of the base claim, but neither Montero nor Wikstrom explicitly teaches that the manager service executes on one of the application servers. However, the peer-to-peer locking system of Devarakonda teaches a distributed locking manager wherein all devices accessing the locked data run a copy of the manager service **[See Devarakonda, Column 3, lines 9-20]**. Thus, the combination of Devarakonda and Wikstrom as discussed above with regard to claim 5 would necessarily run LLMs and LCSs on the application servers.

It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to further modify the local lock manager of Montero-Wikstrom with the distributed lock manager of Devarakonda, with the motivation of reducing the overhead associated with obtaining a lock **[See Wikstrom, Column 2, lines 36-49]** Thus, claim 5 is rejected.

37. Claims **2, 3, and 23** are rejected under 35 U.S.C. 103(a) as being unpatentable over Devarakonda in view of Recovery in the Calypso Filesystem, by Devarakonda, Kish, and Mohindra (hereinafter 'Kish').

38. With regard to **claim 2**, Devarakonda teaches the substantive limitations of the base claim, but is silent on a backup server. However, Kish, in the same field of endeavor, teaches:

a backup computer coupled to the server computer, wherein the backup computer includes a second memory storing program instructions and a second processor coupled to the second memory; wherein the second processor is operable to execute program instructions stored in the second memory to implement a backup manager service **[See Kish, Page 292, Figure 1; and Page 298, Paragraph 2]**;

wherein the manager service is operable to maintain state information regarding which client processes hold which access rights to which data objects **[See Kish, Page 292, Figure 1; and Page 298, Paragraphs 2 and 3]**;

wherein the manager service is operable to communicate with the backup manager service **[See Kish, Page 292, Figure 1]**

wherein the backup manager service is operable to maintain a mirror of the state information maintained by the manager service **[See Kish, Page 298, Paragraphs 2, 3, and 4: The recovery protocol establishes a mirror]**

It would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to include a backup system as described by Kish in the distributed locking system of Devarakonda, with the motivation of providing transpartent fault-tolerence **[See Kish, Page 289, Paragraph 2]**. Thus, claim 2 is rejected.

39. With regard to **claim 3**, Devarakonda further teaches:

in the event that the manager service becomes inaccessible, the backup manager service is operable to become the manager service **[See Kish, Page 298, Paragraphs 2, 3, and 4]**.

40. With regard to **claim 23**, it is a broader version of claim 2, and is rejected for similar reasons.

## *Conclusion*

41. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Concurrency Service Specification, Published in April 2000 by the Object Management Group teaches a similar locking mechanism for CORBA objects.

42. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael R. Gayeski whose telephone number is 571-272-0978. The examiner can normally be reached on M-F: 8:00AM-4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Wiley can be reached on 571-272-3923. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael R Gayeski
Examiner
Art Unit 2143

mrg
2 December 2005

DAVID WILEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100